



Log 3000 – Processus du génie logiciel

Hiver 2021

Examen Final

Présenté par
Andi Podgorica

1955913

28 avril 2021

École Polytechnique de Montréal

Département de Génie Logiciel et Informatique

Code d'honneur En répondant à cet examen à domicile ("take-home") de LOG3000, je m'engage à :

- avoir complété cet examen par moi-même, sans l'aide d'autres personnes;
- ne remettre que mes propres travaux, qui ne contiennent aucun matériel plagié;
- ne pas divulguer les questions posées dans cet examen et les réponses aux questions à quiconque ne suit pas LOG3000 cette session;
- ne pas participer à des activités malhonnêtes visant à améliorer mes résultats ou à nuire aux résultats d'autres personnes.

Signature : *Andi Pedgerica*

Date et lieu : Montréal, le 28 avril 2021

1. Gestion de configuration et de changement (5 points)

(a) Donnez et expliquez 3 arguments pertinents et majeurs à l'équipe de gestionnaires contre la décision de migration totale vers Dropbox. (3 points)

Tout d'abord, une migration totale de Git vers Dropbox n'est pas une bonne idée. En premier lieu, Git permet une meilleure gestion des configurations, c'est-à-dire que Git permet de travailler sur plusieurs fonctionnalités en même temps, où chacun travaille sur sa propre branche. Ainsi, il est plus facile de voir si quelqu'un a introduit une erreur et il y a la possibilité de revenir vers l'arrière pour corriger les erreurs. Contrairement à Dropbox, il est possible de voir le commit de la personne qui a push et modifié le code nous permettant de voir c'est à quel moment que les modifications ont brisé les fonctionnalités. Alors, Git permet de faire une meilleure gestion des changements apportés aux fonctionnalités ainsi qu'aux artefacts. Git fournit également plus de métadonnées permettant de mieux documenter les changements comparativement à Dropbox. En second lieu, Dropbox ne permet pas le contrôle de qualité (code review), c'est-à-dire de voir si les personnes qui ont apportés les modifications ont écrit du bon code ou du mauvais code. En effet, en utilisant Git il est possible de faire de la révision de code pour détecter les odeurs dans le code (merge request). De plus, il est également possible de voir si les nouvelles modifications respectent l'architecture. L'application Dropbox ne permet pas un contrôle de qualité ce qui peut amener à une perte de temps et d'argent à la compagnie si une erreur est introduite. En troisième lieu, Git permet une meilleure communication entre les outils, c'est-à-dire l'intégration avec d'autres technologies. En utilisant Git, il est beaucoup plus facile de faire la gestion de plusieurs outils en même temps. Ceci est une pratique que beaucoup d'entreprises utilisent et l'intégration d'outil de travail comme Gerrit à Git par exemple, va permettre une amélioration ainsi qu'une facilitation de la gestion de travail.

(b) Expliquez, avec des scénarios concrets, comment des problèmes de traçabilité peuvent impacter les processus de développement. (1 point)

D'abord, la traçabilité nous permet de retracer les modifications apportées sur un artefact. À l'aide de la traçabilité, on peut connaître la personne qui a effectué les changements, la raison de ces changements ainsi que la date des modifications. Ainsi, il est facile de documenter toutes les modifications. Alors, lorsqu'il y a des problèmes de traçabilités, il est difficile de savoir à quel moment et quelle personne a apporté des mauvaises modifications à l'artefact. Donc, ceci peut augmenter la tâche des personnes en essayant de trouver les modifications apportées et génère une grosse perte de temps. De plus, il y a un manque de liaison entre les outils utilisés par la compagnie. Ceci peut générer des problèmes de traçabilité. Par exemple, lorsqu'une personne ne prend pas une tâche sur la plateforme Jira, d'autres personnes vont s'assigner cette tâche provoquant ainsi un temps de travail mal réparti. Aussi, lorsque les commit sont mal écrits sur Git, il est difficile de savoir sur quelle fonctionnalité une personne a travaillé et il est

également difficile de savoir quelle fonctionnalité amène des bogues. Ceci compliquerait la tâche de l'équipe car il est compliqué et long de retrouver la source du problème. Alors, le processus de développement sera retardé.

(c) Proposez des solutions pour améliorer la traçabilité dans le processus de « Connectons-nous ». (1 point)

Une des solutions proposées est d'identifier les personnes travaillant sur les artefacts dans les différents outils. Il est également nécessaire de s'assigner des tâches sur Jira afin d'indiquer sur quelle tâche une personne travaille afin de permettre une meilleure traçabilité en cas de problème. De plus, il est nécessaire de bien expliciter les *commit* sur Git avec des commentaires précis afin de tracer le *commit* causant des problèmes dans le code au besoin. Bien documenter les fonctionnalités sur le Wiki est une autre solution pour améliorer la traçabilité. Enfin, une dernière solution proposée est de travailler sur un outil commun. En effet, le fait de travailler sur un outil commun permet une plus facile traçabilité en cas d'introduction d'erreur et permet d'éviter les incohérences entre différentes plateformes.

2. Principes de DevOps (7 points)

(a) En utilisant un exemple concret, expliquez la relation entre les requis des développeurs et les requis des opérateurs. (2 points)

Tout d'abord, l'équipe de développeurs permet de mettre en place un plan, du code fonctionnel, le « build » de ce code ainsi que les tests de fonctionnalités. Le but des développeurs est de concevoir un produit final qui pourrait être utilisé par les clients. À l'opposé, l'équipe des opérateurs se concentre sur l'acquisition de stocks et des services maximisant la productivité, minimisant les coûts et répondant aux attentes des clients. Ils se concentrent particulièrement sur la diffusion de ce que l'équipe de développement leur a fourni, le déploiement, l'exploitation et l'affichage de l'application. DevOps est une approche organisationnelle permettant la collaboration inter fonctionnelle entre les équipes, plus précisément le développement et les opérations informatiques dans des organisations de développement logiciels. Le but est d'exploiter des systèmes résilients et d'augmenter la mise en œuvre des changements. De plus, DevOps vise à réduire le temps entre un changement dans un système et le changement et la mise en production de la modification tout en garantissant une grande qualité. Enfin, DevOps permet de partager des informations entre les deux équipes afin d'augmenter l'efficacité et la responsabilité partagée.

- (b) Donnez et expliquez 3 arguments pertinents et majeurs à l'équipe de gestionnaires pour d'adopter DevOps et ses principes. Dans la réponse, vous pouvez faire des hypothèses sur l'entreprise, mais vous devez être explicite sur vos hypothèses. (3 points)**

Un des avantages de DevOps est l'harmonisation entre l'environnement de développement et celui de production. En effet, DevOps permet une agréable atmosphère de travail motivant ainsi les équipes. L'enthousiasme dans la compagnie permet d'augmenter en productivité et en qualité de notre produit, mais aussi en efficacité.

Ensuite, un autre avantage de DevOps est la réaction plus rapide aux bogues rapportés. En effet, on passe moins de temps à trouver des solutions aux bogues permettant ainsi au développement et à la mise en marché du produit à ne pas être reporté. Ceci permet une bonne prestation devant les consommateurs et les investisseurs des autres compagnies.

Enfin, un avantage d'adopter DevOps est la mise en production et la maintenance plus rapide d'une application. Effectivement, il est plus facile de mettre en marché notre produit et il est plus facile de le mettre à jour ou d'apporter des corrections.

- (c) Donnez votre opinion sur cette vision de l'équipe de gestionnaires. (2 points)**

Mon opinion sur cette vision de l'équipe de gestionnaires est que l'embauche d'un ingénieur DevOps pour créer une chaîne d'outils automatisée n'est pas idéale car il faut prendre en considération que les deux équipes n'ont pas travaillé ensemble. En effet, les développeurs doivent apprendre des concepts et des langages de programmation des opérateurs. De plus, des problèmes techniques peuvent venir affectant la sécurité et la fiabilité des systèmes critiques à la suite de la fusion de ces deux équipes. L'apport de ces changements peut être coûteux en temps et en ressources.

3. Gestion d'environnement

- (a) Expliquez et comparez les trois modèles de services suivants dans le contexte d'environnement de nuage : SaaS, PaaS, IaaS. (2 points)**

SaaS : Le logiciel en tant que service (SaaS) est un modèle de distribution de logiciels où un fournisseur de cloud héberge des applications et les met disponibles aux utilisateurs sur internet. En bref, le SaaS a des fichiers de configuration ainsi que des données déployées dans un nuage contenant une application avec des bibliothèques et l'OS disponibles aux internautes. Le SaaS utilise donc internet pour fournir des applications aux utilisateurs et ces applications s'exécutent à l'aide de leur navigateur web, ce qui ne nécessite aucun téléchargement ou installation des clients. Un des avantages est la réduction du temps et de l'argent consacrés aux tâches comme l'installation, la gestion ainsi que la mise à niveau.

des logiciels. Le personnel technique a donc plus de temps et peut se concentrer à des tâches plus urgentes.

Paas : Le (Paas) fournit des composants en nuage à certains logiciels utilisés principalement pour des applications. Le Paas fournit aux développeurs un cadre sur lequel ils peuvent se baser et qu'ils peuvent utiliser pour créer des applications personnalisées. En bref, le Paas a des applications et des données déployées dans un nuage contenant des librairies et l'OS. L'un des avantages du Paas est le développement et le déploiement d'applications simples et rentables. De plus, les développeurs peuvent personnaliser les applications sans avoir à faire de la maintenance du logiciel. Ceci réduit donc la quantité de code à faire.

IaaS : L'infrastructure en tant que service (IaaS) est un service de « cloud computing » où des entreprises font la location des serveurs pour le calcul et le stockage dans le cloud. L'IaaS permet d'éliminer le besoin d'approvisionner et de gérer manuellement des serveurs physiques dans les centres de données. L'IaaS permet l'accès et la surveillance des ordinateurs, des réseaux et d'autres services en libre-accès. En bref, l'IaaS est la composition d'une application, des librairies, des données ainsi que l'OS qui sont déployés dans un nuage. L'un des avantages est le déploiement facile et automatisé du stockage, des réseaux et des serveurs. De plus, les clients ont un contrôle total de leur infrastructure.

Similarités : Les trois modèles de services permettent de mettre des ressources sur le cloud.

Différences : Contrairement au SaaS, le Paas fournit une plateforme fournie via le web pour la création de logiciels. Ainsi, les développeurs ont l'opportunité de se concentrer sur la création de logiciels sans avoir à se soucier des systèmes d'exploitation, des mises à jour ou du stockage de l'infrastructure. Une différence que le SaaS présente contrairement aux autres est que l'application est dans le nuage pour le SaaS. À l'opposé, l'IaaS est le seul modèle de service qui a l'application, les librairies, les données et l'OS localement.

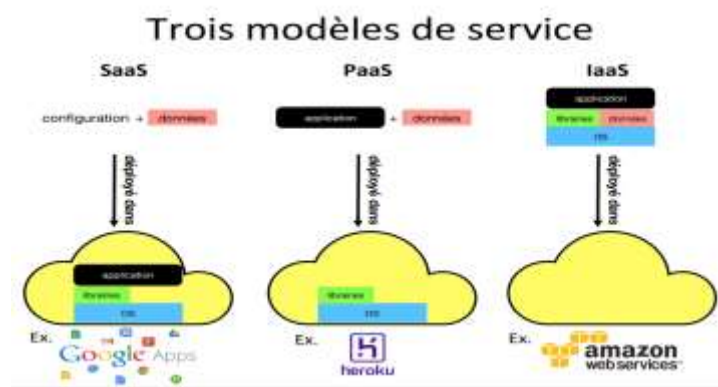


Figure 1. Trois modèles de service

- (b) Identifiez et expliquez de façon détaillée l'environnement requis pour déployer l'application « Connectons-nous », un Software as a Service (SaaS) similaire à Zoom et WebEx, qui prend en charge la vidéoconférence et la collaboration à distance. (2 points)**

L'environnement requis pour déployer l'application serait déployée dans le nuage. Le nuage contient l'application, les librairies et l'OS. Les utilisateurs peuvent donc se connecter à l'application sur internet dans leur navigateur web en ne nécessitant aucun téléchargement ou installation.

- (c) Expliquez aux gestionnaires les avantages et les désavantages d'utiliser une technologie de conteneurs comme Docker en comparaison avec une technologie de machines virtuelles. (2 points)**

Le principal avantage d'un conteneur est qu'il prend moins d'espace mémoire qu'une machine virtuelle. En effet, les conteneurs ne possèdent pas la partie OS, qui est le système d'exploitation, dont nous retrouvons dans les machines virtuelles. Aussi, les conteneurs sont plus rapides car ils ont un déploiement ainsi qu'un chargement des applications plus rapides que celle d'une machine virtuelle. De plus, il est possible de virtualiser un système d'exploitation avec des conteneurs afin que les charges de travail puissent fonctionner sur une seule instance de système d'exploitation. À l'opposé, le matériel des machines virtuelles est virtualisé permettant d'exécuter plusieurs instances de système d'exploitation. Un des désavantages des conteneurs est que l'isolation au niveau du processus est moins sûre. Un autre désavantage d'utiliser des conteneurs est que toutes les applications ne bénéficient pas des conteneurs. En effet, seulement les applications conçues pour fonctionner comme un ensemble de micro-services discrets ont l'avantage d'utiliser des conteneurs.

4. Déploiement et mise en production

- (a) Identifiez et expliquez les différences entre les deux concepts : déploiement et mise en production. (1 point)**

Tout d'abord, le déploiement est une version du logiciel qui est déployée une fois qu'elle est installée et configurée dans l'infrastructure, mais pas encore accessible aux clients ou aux utilisateurs. En bref, le déploiement veut dire que le logiciel est installé dans l'infrastructure de production. À l'opposé, la mise en production est l'exposition de ce logiciel aux utilisateurs, c'est-à-dire sa mise à disposition. Donc, la mise en production veut dire que le logiciel est disponible sur le marché et il est possible aux utilisateurs d'y avoir accès.

(b) Expliquez pourquoi il est important de faire la distinction entre les processus de déploiement et de mise en production. (1 point)

Il est important de faire la distinction entre les processus de déploiement et de mise en production afin de découpler les deux processus, c'est-à-dire de séparer ces deux étapes en sachant que le déploiement renvoie au processus qui consiste à l'intégration dans l'environnement de production réel et la mise en production désigne le développement d'une nouvelle version d'un service ou d'un composant. Enfin, le déploiement n'affecte pas les utilisateurs (découplage), mais la mise en production affecte directement les utilisateurs. La mise en production peut être découplée du déploiement. Il est donc important de ne pas livrer un produit qui n'est prêt.

(c) Décrivez et expliquez brièvement des techniques que vous pouvez adopter pour résoudre ce problème. Astuce : une illustration visuelle pourrait aider à soutenir votre explication des concepts pertinents. (3 points)

Une solution que je propose est d'utiliser le concept de « release trains ». Le but de ce dernier est de réduire le *cycle time* afin de faire plus de *releases*. Ainsi, le temps pour une *release* est plus beaucoup grand que celui d'un *cycle time* en sachant qu'il y aura une autre *release* après ce dernier. Ainsi, si le travail n'est pas terminé, il y aura une autre *release* qui viendra permettant d'inclure la correction des bogues ainsi que les nouvelles fonctionnalités.

(d) Quelle stratégie de mise en production proposeriez-vous ? Pour la stratégie choisie, expliquez en détail aux services informatiques comment ils devraient implémenter une telle stratégie, quelles mesures sont nécessaires à surveiller, etc. (3 points)

Je propose la stratégie de A/B testing. En effet, pour que les services informatiques implémentent cette stratégie, il faut déployer les deux versions de l'app. Les services informatiques doivent également répartir le trafic d'utilisateur pour les guider vers les deux versions. De plus, les services informatiques doivent se baser sur les données de l'expérience des utilisateurs ainsi que sur les données sur la performance de l'application pour les deux versions. Enfin, on doit faire la comparaison de ces données pour finalement déployer la meilleure application possédant les meilleures données. On doit également diriger le trafic vers cette dernière version.

5. Intégration

(a) Expliquez le concept de « portes de qualité » en termes simples pour que l'équipe de gestionnaires sans formation technique puisse le comprendre. (1 point)

Les portes de qualité sont un sens technique de l'intégration continue. En effet, il s'agit de fusionner fréquemment les modifications de code dans un dépôt central où les constructions et les tests sont ensuite exécutés. Cette pratique exige les développeurs à

intégrer du code dans un référentiel partagé plusieurs fois par jour. Ensuite, il y a une vérification permettant aux équipes de détecter rapidement les problèmes.

(b) Comparez et expliquez l'approche manuelle et l'approche automatique de « portes de qualité ». Et puis, décrivez et justifiez l'approche que vous allez adopter chez « Connectons-nous ». (2 points)

L'approche manuelle de portes de qualité consiste en une révision technique. Premièrement, cette révision technique est une activité collaborative concernant l'inspection manuelle d'un changement de code qui est supportée par des outils web tels que GitHub pull request. Ensuite, il faut suivre une politique spécifique pour les directives de programmation à suivre et il faut également voter l'acceptation du patch lorsque le patch est intégré en étant un commit dans le référentiel Git du projet. Enfin, la révision technique de la porte de qualité manuelle demande une collaboration entre les développeurs de l'application et les réviseurs (reviewer) dans le but d'améliorer le patch de départ de façon constructive jusqu'à l'acceptation de ce dernier. À l'opposé, l'approche automatique de portes de qualité consiste en une approche de vérification automatique qui peut être enclenché par Git ou par le web et que cet enclenchement peut être programmé à des heures précises. L'enclenchement de la vérification programmé s'assure de la mise en marche de l'application. Alors, à chaque fois que quelqu'un push du code et selon le temps de l'enclenchement, le système va vérifier si le code qu'on a push ne brise pas l'application. L'approche à privilégier pour « Connectons-nous » est l'approche manuelle car on doit contrôler la qualité du code soumis qui doit être faite par des humains. En effet, ni Git, ni le web ne peuvent s'assurer de la qualité de code qui a été soumise, donc c'est pour cette raison que l'approche automatique est à négliger.

(c) Énumérez et expliquez 3 pratiques que vous demanderez aux contributeurs externes de suivre pour éviter les conflits lors de l'intégration. (3 points)

Les 3 pratiques demandés aux contributeurs externes seraient :

1. Ils doivent faire des commits fréquemment afin d'éviter les conflits lors de l'intégration. Le fait de faire des commits fréquemment donne l'avantage de trouver les commits qui ont cassé quelque chose dans l'application et on est capable de revenir en arrière sans perdre notre travail. De plus, des commits fréquents évitent de mélanger plusieurs changements non-liés.
2. Ils ne doivent pas faire de commit si leur code ne compile pas ou ne passe pas les tests. En effet, si on fait un commit quand le code ne compile pas, il est difficile de retrouver dans quel fichier qu'on a modifié l'erreur se trouve. Les tests doivent passer car sinon il y aura des erreurs dans le système d'intégration continue.

3. Ils doivent corriger directement le code si l'intégration continue casse car personne dans l'organisation pourra compiler ou tester le système. En effet, du code qui ne fonctionne pas bloque tout le processus d'intégration et de tests.

6. Surveillance des opérations

(a) Expliquez le concept de « ingénierie du chaos » et pourquoi (ou pas) cette pratique peut améliorer la fiabilité d'une application. (2 points)

L'ingénierie du chaos est une multitude d'expérimentations sur un système distribué dont le but est de renforcer la confiance dans la capacité de système à résister à des conditions turbulentes en production. En d'autres mots, l'ingénierie du chaos consiste au fait de faire des tests en essayant de casser le système de production afin de trouver des failles. Cette pratique peut améliorer la fiabilité d'une application car elle teste tous les cas possibles pouvant créer une défaillance dans notre système de production. Ainsi, on s'assure de façon proactive pour détecter comment le système utilisé en production réagit sur des pannes.

(b) Expliquez comment, concrètement, « Connectons-nous » pourrait implémenter la pratique de « ingénierie du chaos ». Dans la réponse, vous pouvez faire des hypothèses sur l'entreprise et/ou l'application, mais vous devez être explicite sur vos hypothèses. (2 points)

« Connectons-nous » pourrait implémenter la pratique de « l'ingénierie du chaos » en mettant en place des pratiques de surveillance. Voici quelques pratiques de surveillance utilisés :

1. Il faut mettre en place une surveillance permettant de tout surveiller. En effet, il faut surveiller l'infrastructure en regardant l'utilisation du CPU, en regardant le temps de réponse, le stockage et le réseau. Aussi, il faut surveiller les performances des applications, c'est-à-dire l'analyse des transactions et le profilage. Enfin, une surveillance de bout en bout surveille également les activités de l'utilisateur, dont la fréquence de ses connexions et les utilisations des fonctionnalités.
2. Il faut mettre en place des alertes lorsque les métriques changent par rapport à la référence. En effet, les métriques (comme des mesures de surveillance de l'infrastructure, des mesures des incidents de haute gravité) permettent de détecter si on respecte les normes et lorsqu'on ne les respecte pas, on reçoit des alertes.
3. Une autre bonne pratique est de gérer le tableau de bord pour afficher les informations clés dans une vue holistique.

7. Réflexion

- (a) Le fait que la plupart des cours sont donnés à distance a probablement eu des impacts majeurs sur le processus de votre projet intégrateur et / ou de vos travaux pratiques. Réfléchissez à ces impacts et écrivez un bref « post-mortem » à l'un de vos projets ou TP. Dans le « post-mortem », examinez les aspects tels que les difficultés spécifiques que vous avez rencontrées, la façon dont vous les avez gérées, ce qui s'est bien passé, ce qui n'a pas fonctionné, ce que vous pourriez faire différemment la prochaine fois dans une situation similaire, etc. (3 points)

Un des impacts majeurs des cours à distance, principalement pour le tp4 de log3000, est le fait que je n'ai pas été motivé à poser des questions lorsque je ne comprenais pas. En effet, lorsqu'on pouvait assister en personne aux séances de laboratoire, je posais toujours des questions aux chargés de laboratoire quand je ne comprenais pas comment faire les démarches du TP. Cependant, à distance, je n'étais pas très motivé à poser des questions aux chargés. En effet, il est moins motivant pour moi d'assister aux séances de laboratoire en ligne. Donc, lorsque je n'arrivais pas à répondre à une question, je regardais le plus d'articles en ligne pour s'informer sur le sujet. Ceci a été très productif car j'ai appris non seulement à bien répondre aux questions de laboratoire, mais aussi à assimiler et mieux comprendre la matière. Par contre, cette approche est plus longue et prends plus de temps. Ce que je ferai de différent la prochaine fois est de poser plus de questions aux chargés de laboratoire même lorsque j'ai accès à beaucoup d'outils et d'informations en ligne.

- (b) Donnez 3 suggestions pour améliorer le cours LOG3000. (1 point)

1. Les diapos devraient avoir plus d'informations sur les concepts de base, c'est-à-dire qu'il faut plus détailler et donner des exemples concrets pour nous permettre de comprendre la matière.
2. Des exercices faites par le prof auraient contribué à l'assimilation de la matière.
3. Les images des PowerPoint devraient être plus claires, présentant des informations détaillées en lien avec la matière.
4. Diminution de la longueur des examens « take home », car en réalité on dépasse toujours de beaucoup le temps qu'on consacre par rapport au temps qui nous est communiqué.

Je vous souhaite des belles vacances et profitez-en bien 😎😊